RIT Computer Science MS Project / Thesis Proposal

Title: A dynamic web server based appointment calendar

Name: Jianxia Zhong

Abstract

Nowadays the Internet has become the most popular way for people to sharing their information. While many of the web pages on the Internet are static web pages, they don't change in response to user input. With the development of the server side technology, more and more web application allows users to interact with the web server. This is the dynamic web application.

In this project, I will use the most advanced server side technology, Microsoft .NET Framework, which is running on the Windows operating system built with IIS (Internet Information Services) and Microsoft SQL Server 2000, which is used as the Database server for the persistence layer to develop a dynamic web server based calendar to handle appointment and meeting scheduling. It's a new calendar in architecture and functions comparing to the traditional calendars. The existing commercial calendars, such as Outlook calendar, iCal and MeetingMaker, have a common weakness: unique calendar software should be installed in user's computer. They are all not web based calendar. Some existing web calendar, like Yahoo calendar, their function is very limited. For example, they couldn't get the response from the meeting attendee automatically and view the attendee availability. In this proposal, a web server based calendar is proposed to overcome this weakness, i.e., no unique software is needed and with lots of very useful function. The user can access the calendar service wherever a web browser exists. A comparison analysis will be done between the proposed web server based calendar and those existing ones, focusing on their functionality, effectives and User Interface with look and feel. This web calendar will be built with the functions like creating appointment, proposing a meeting, viewing the attendee's availability and the responses, accessing and even changing the calendars of other users based on the given permission.

Introduction

1. The introduction of the existing calendars

In today's world, people rely on email and e-calendar more and more. It happens not only in companies, universities, and government departments, but also in all kinds of societies and organizations. It's very convenient to track appointments and tasks by using e-calendar. It's especially efficient in scheduling meetings as online access of the availability of the attendees and resources is made possible. The responses from the attendees update the meeting attendance table automatically to make the meeting organizer know who are planning to attend the meeting. The reminder function of the e-calendars prevents the users from overlooking any events.

Several commercial e-calendars are available, such as Outlook calendar, iCal and MeetingMaker. These traditional calendars are widely used today. Although it makes easy to set appointments and meetings, these calendars require unique software be installed in user's computers. It's sometimes very inconvenient, such as in traveling and remote accessing.

Outlook calendar [1] is an application of Microsoft Office Suite. It has the largest user population because over 90% of personal computers have Microsoft Office Suite as the major office software. Both the Outlook exchange server and clients are running in local intranet.

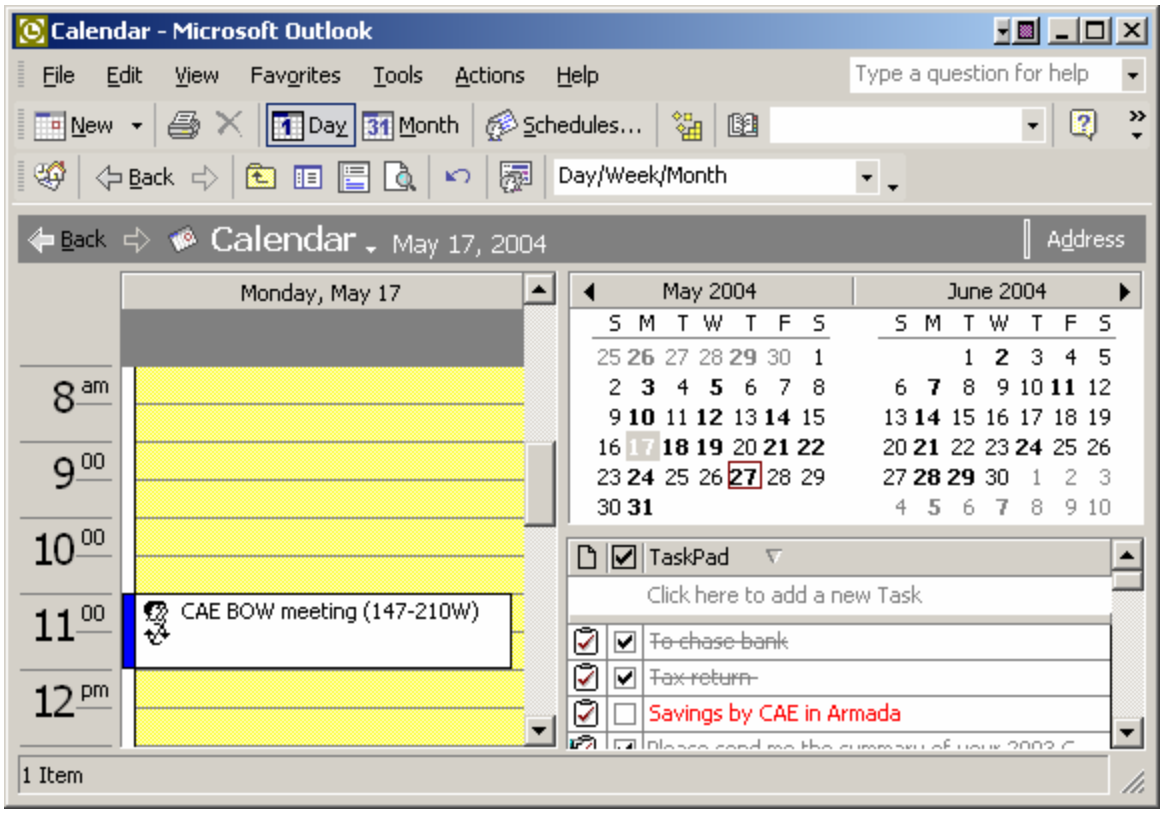


Fig.1 Outlook calendar in Day View layout

Fig.1 shows a typical UI of Outlook calendar. In Outlook calendar, a user can set appointments, propose meetings, set reminders, check the availability of others, check the responses of meeting requests, … Fig.2 schematically shows how Outlook calendar works.

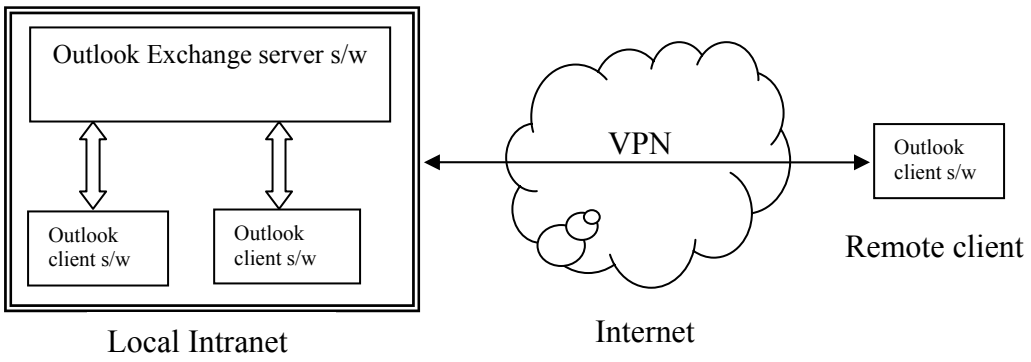


Fig.2 Illustration of how Outlook calendar works

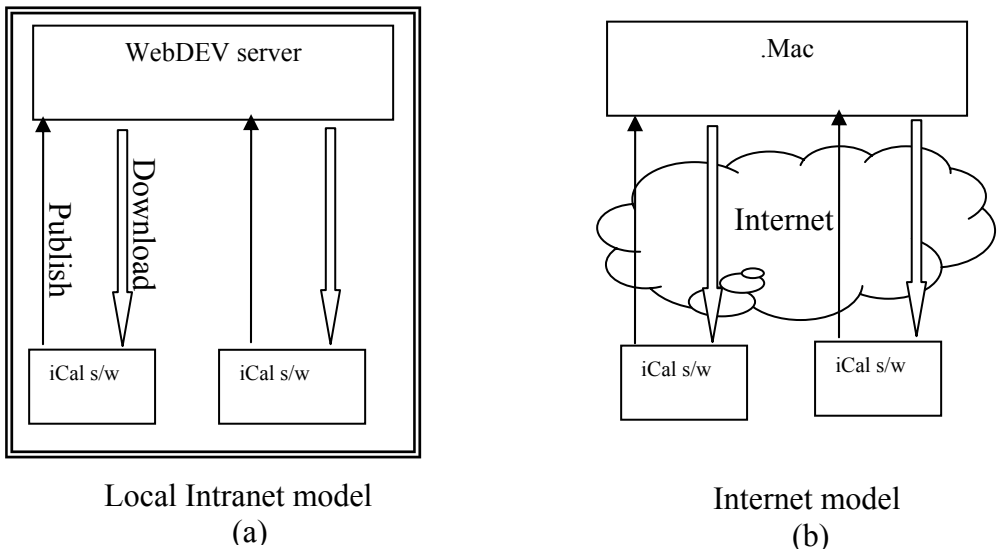
An Outlook exchange server is running in the local intranet, and Outlook client software is running in the client computers. The clients and the server have interactions to exchange information. A remote Outlook client can also exchange information with the server through Internet by Virtual Private Network (VPN). VPN software must also be installed in the remote client to access the exchange server.

While Outlook Calendar running in Microsoft Windows platform, iCal calendar is running in Macintosh OS platform [2]. Even through their functionality looks similar, they are built on different architectures. Fig.3 shows a typical UI of iCal calendar. In iCal, different calendars are color coded and displayed all together. The calendars of other users can also be displayed together with the user's own calendars. It has similar functions as those of Outlook, i.e., setting appointments, meetings, inviting attendees, setting reminders. iCal is a stand-alone application that must be installed in the local computer. A user's calendars can not be accessed by other users unless they are published on the web. That means others don't know when you are available unless you publish your calendars in the web. iCal publishes calendars on .Mac or local WebDEV server, so others can access those calendars using web browsers. If they are iCal users, they can even subscribe to your calendar and have immediate access to it from iCal.

Fig.3 iCal Calendar in Day View layout

Fig.4 shows two working models of iCal to share calendars. iCal publishes calendars on a local WebDEV server, so the users in the local intranet can share the

published calendars, as shown in Fig.4(a). iCal publishes calendars on .Mac through internet, so users on the internet can access those published calendars, as shown in Fig.4(b).



Local Intranet model
(a)

Internet model
(b)

Fig.4 iCal sharing calendar models

When a user requests a shared calendar using web browser, the .Mac or local WebDEV server delivers the shared calendar to the client's web browser. But the interaction between the web browser and the .Mac or WebDEV server is very limited.

MeetingMaker is copyrighted by Meeting Maker Inc., which is running in cross platforms, such as Windows, Macintosh, and Solaris [3]. The secure communication across the internet between the clients and server is through SSL (Security Sockets Layer) protocol. Fig.5 schematically shows the working model of MeetingMaker.

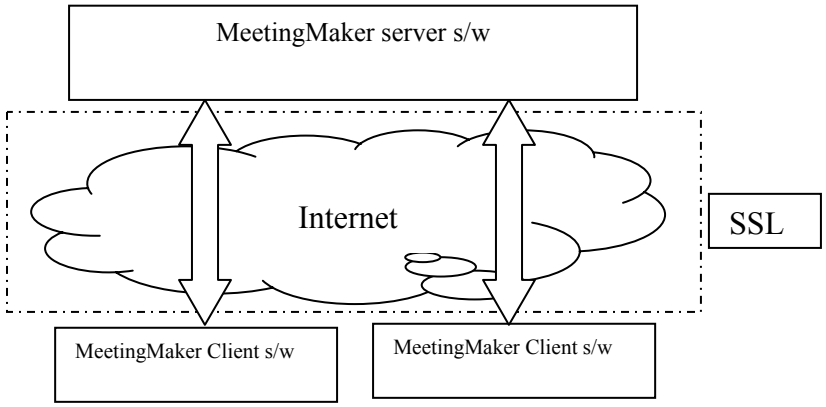


Fig.5 Illustration of how MeetingMaker works

The MeetingMaker client software must be installed in each client computers. The client computers can be a mixture of different OS platforms. A MeetingMaker server is running to provide service to the clients. The clients communicate with the server interactively through the internet. The SSL protocol is used to provide secure communication. MeetingMaker can also publish calendars on the web, so other users can access the shared calendars using any web browsers. Again the interaction between MeetingMaker server and the client's web browser is very limited.

Yahoo calendar [4] is a web server calendar too, which doesn't require unique client software and can be accessed wherever a web browser exists. But its functionality is very limited. For instance, the server can not take the response from the meeting attendees automatically so that the meeting organizer doesn't know who accept and who reject the meeting request; the meeting proposer can not check the availability of the attendees. These two shortages keep Yahoo calendar off the effective meeting scheduling tools. Yahoo calendar is too simple to be used in the places that functions and efficiency are required. There is also security concern that the packages transported between Yahoo calendar's web server and the user browser are not encrypted.

2. The introduction of the proposed calendar and the comparison with those existing calendars.

A new calendar model is proposed in this proposal to overcome the weakness of the existing calendars. The calendar web application will be running at a web server and the users only need a web browser to access it if the user is granted the permission to access it. The web server based calendar application can 'percept' the client behaviors and take correspondent actions, i.e., the server will send dynamic web pages to the clients according to the clients' behaviors. Fig.6 shows the working model of the proposed dynamic web server based calendar. It's a two-way traffic model of communication between the server and the clients.

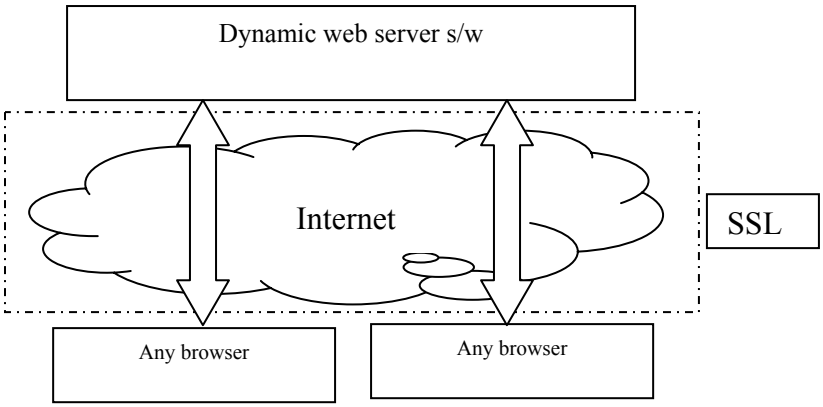


Fig.6 Illustration of the dynamic web calendar model

This dynamic web server based calendar will have unmatched conveniences since it can be accessed wherever there is a web browser. It will also have the powerful appointment and meeting scheduling capacities. The proposed dynamic web server based calendar is compared with other calendars and the results are listed in table 1.

Table 1: Comparison of different calendars

	Outlook	iCal	MeetingMaker	Yahoo	Proposed
Without unique client software	-	-	-	+	+
Dynamic web pages (iCal and MeetingMaker publish calendars on the web as static pages)	n/a	-	-	+	+
Secure communication	+	+	+	-	+
Information stored in server only	+	n/a	-	+	+
Meeting status updated by attendee response automatically	+	-	+	-	+
Check attendees' availability when proposing meetings	+	-	+	-	+
Meeting list	-	-	+	-	+
Windows look and feel UI	+	n/a	+	-	+

As discussed before, only the proposed calendar and Yahoo calendar don't need unique client software.

iCal and MeetingMaker also use web to publish calendars, but the published calendars are read only (can not be modified) when accessed by web browsers.

Yahoo calendar doesn't use secure communication in the internet, i.e., the packages transported between Yahoo server and the clients' web browsers are not encrypted.

iCal stores calendar information (appointments, meetings,…) in local Apple computer. Other users can not access it before it is published in the web. Meetingmaker stores calendar in both client and server. The calendars stored in client and server synchronize periodically. This model of date storage may cause problem sometimes when a PDA tries to synchronize calendar in different computers but the computers haven't synchronized with the server.

iCal and Yahoo can not update the meeting status automatically, so the meeting organizer doesn't know who is going to attend the proposed meeting and who won't. iCal and Yahoo calendars don't enable the meeting planers to check the attendee's availability too. These two major disadvantages keep iCal and yahoo off effective meeting planning tools.

Only the proposed dynamic web server calendar and MeetingMaker have 'message list' to list the meetings for easy assessment. This feature is especially useful when a user has many meetings weekly for it provides an index of meetings so the user doesn't need to look for them all over in the calendar.

The proposed calendar, Outlook and MeetingMaker have the so called 'Windows look and feel' user interface, which make them very user friendly since many people are used to the Windows UI.

The proposed dynamic web server based calendar is the only one to have all the listed features or advantages. The advanced technology of .NET framework makes it possible that will be discussed in the section of design specifications.

Till now, there is not a mature dynamic web server based calendar on the marketplace. The proposed calendar in this project will be the first dynamic web server based calendar developed in .NET framework that can be used by end users.

3. The introduction of the technologies used in this project.

To implement the dynamic web server based calendar model in Fig.6, ASP.NET is the right tool for creating rich and dynamic Web sites. Prior to the advent of ASP.NET, three main technologies and platforms were available to develop web applications: ASP, Java Server Pages(JSP), and the Open Source web platform commonly referred to as LAMP (Linux Apache plus MySQL plus either perl, Python, or PHP as the programming language. But the ASP.NET is currently the most technologically advanced, feature-rich, and powerful platform for building distributed applications transported by the HTTP protocol. It provides a new programming model and infrastructure that provides a component based, extensible and easy-to-use way to build, deploy, and run web applications that target any browser [5]. ASP.NET is a compiled .NET Framework-based environment where applications can be authored in any .NET Framework compatible language including Visual Basic, C#, J#, C++, and JScript. So unlike classic ASP, in which you had to depend on interpreted VBScript and JScrip languages, ASP.NET provides a wide choice of programming languages.

In addition to the usual web applications, ASP.NET allows developers to create other types applications, which enables you to extend your applications' reach to new customers and business partners. For example, XML Web Services enable sharing of data across the Internet regardless of the operating system and the programming language,

also ASP.NET provides lots mobile controls that enable your application to target a large number of mobile web devices.

Besides the advantages mentioned above, the .NET Framework also provides another benefit to developers: Direct Support for Security. Window NT, Windows 2000, and Windows XP have a number of security features based on Access Control Lists (ACLs). An ACL contains a number of entries that specify which users may access, or are explicitly denied access, to resources such as files and printers. ACLs are a great way of protecting executable files (applications) from unauthorized access. But they do not secure all parts of the file. The .NET Framework enables both developers and system administrators to specify method-level security. The .NET Framework uses industry-standard protocols such as TCP/IP and means of Communications such as the Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), and HTTP to facilitate distributed application communications. This makes distributed computing more secure.

ASP.NET uses the Common Language Runtime (CLR) provided by the .NET Framework. The CLR is the .NET runtime, which manages the execution of code. The CLR allows the objects, which are created in different language, to interact with each other and hence removes the language barrier. CLR thus makes web application development more efficient.

Microsoft made most ActiveX controls be available in the ASP.NET [6], which enabled a website authored in ASP.NET could deliver similar functionality as a Windows application. The ASP.NET supports several kinds of language, such as Visual Basic, Visual C#, Visual C++, Visual J++, and Visual J# [7]. Basically, any of the language can do the same implementation. Visual Basic and Visual C# are the most popular language among them for they are simple to use. C# .NET will be used as the coding language in this project.

Databases often play an integral part in storing the information, and the scripting language query the database to create the dynamic web page. Some popular relational databases which are used for the persistence layer are Oracle, DB2, Microsoft SQL Server, Sybase, MySQL, Postgre, and MS Access. The choice of technology depends on the target environment. For ASP.NET, the two most popular databases used are

Microsoft SQL Server and Oracle. In this project, Microsoft SQL Server 2000 will be used as the database server.

Functional Specification

Web Calendar Layouts

The web calendar can be displayed in two layouts: By Day and By Month. Appendix p.1 shows the Day layout calendar. The top row of buttons are the task buttons. The first button, Add appointment, will open the appointment scheduling page, as shown in Appendix p.14. The second button, Add meeting, will open the meeting scheduling page, where the user proposes meetings, as shown in Appendix p.4. The third button, Message, will open a page to show the list of incoming meeting requests, as shown in Appendix p.10. The Go to Today button will display today's appointments and meetings. The Contacts button will open a page to display the accessible users for meeting scheduling, as shown in Appendix p.8. The Proxy button will open a page, as shown in Appendix p.2, for giving privilege to other persons to access the user's calendar. The user can also access other person's calendar from here.

The second row of buttons are for the display layout. There are two buttons in this row: By Day and By Month. Clicking on the By Day button will open the day layout calendar. Clicking on the By Month button will open the month layout calendar, as shown in Appendix p.3.

On the right side of the layout buttons, a text box will display today's date.

A calendar control is on the right side of the calendar with the current month displayed by default. The days in the calendar control are clickable. It will open the correspondent calendar of that day by clicking on that day in the calendar control. Three navigation buttons are on the top of the calendar control to let the user to choose any month and year.

The above buttons and features are common to the two layouts. The only difference between the two layouts is the scheduling field. In day layout calendar, this field displays the appointments and meetings of a chosen day, as shown in Appendix p.1. By clicking on the Add appointment or Add meeting button, the appointment or meeting scheduling page will be opened, as shown in Appendix p.4 and p.14. The scheduled

appointments and meetings are displayed in the table as blocks that occupy the first half hour time range of those events. They are clickable for viewing details or for modification. The month layout displays the appointments and meetings of a whole month, each day occupies a table cell, as shown in Appendix p.3. Clicking on the date link in a cell will open the day layout calendar of that correspondent day. The appointments and meetings are listed in the table cells as links. Clicking on a link will open it for viewing and editing.

Management of Proxies

A user can let other users to access his or her own calendar by assigning proxy privilege to them, as shown in the region framed by solid lines in Appendix p.2. Firstly, highlight the person you want to assign proxy in the ListBox. Secondly, click on the 'Add to Proxy ->' button to add the person to the 'Assigned Proxies' ListBox. By default, the assigned proxy is read only, i.e., that person can view your calendar, but cannot modify it. You can change it to 'Read and Write' by clicking on the Read/Write button. The 'Properties' button will open the contact information page of the chosen person, as shown in Appendix p.9, which provides some identity information of other persons to the user. A user can display and/or modify other person's calendar if he or she has the permission. It's illustrated in the region framed by the dash lines in Appendix p.2.

Scheduling Appointments and Meetings

The UIs to schedule appointment and meeting look similar. The difference is that an appointment will not involve other persons, so it doesn't have the tab of Attendee Availability and the To field. The meeting scheduling page is illustrated in Appendix p.4, there are two tabs in the tab strip: Meeting tab and Attendee availability tab. The Meeting tab shows this page (Appendix p.4) and the Attendee availability tab shows another page (Appendix p.6). If the 'Private' check box is checked in the page of Meeting tab, this piece of appointment or meeting cannot be seen by others. By clicking on the To link, the contact page will be opened for the meeting scheduler to invite attendees and reserve resources, as shown in Appendix p.8. An attendee can be marked either as required person or as optional person. The subject and location are not required for the program but are necessary for the user; otherwise, the invited attendees will not know what the

meeting is for. The starting and ending time of the event can be input in this page. The 'Show time as' DropDownList has five options: Free, Tentative, Busy, Out of office and No information. It shows the user's availability in that time period to other meeting schedulers. The user can type or paste appointment notes or meeting notes in the 'Notes' TextBox.

The page of the Attendee availability graphically shows the availability of the meeting attendees, as shown in Appendix p.6. The meeting scheduler is the default first attendee in the 'All attendees' column. Three navigator buttons above the availability table help the organizer to see the availability of the attendees in any day. The meeting scheduler can set the starting and ending time in this page also.

If the link show attendee status was clicked, the page of attendee status will be opened, as shown in Appendix p.7. This is a read only page. It will show the attendee status and responses. During planning a meeting, the 'Response' column is empty. When an invited attendee receives the meeting request, his or her calendar is automatically marked by that meeting request and that meeting request also shows up in the Message list, as shown in Appendix p.10. After a user received a meeting request, he or she can click on the Message button (on the top of the calendar) to open the Message list page. By clicking on the link in the Subject column, a read only meeting page will be opened for the user to response, as shown in Appendix p.11. If the user doesn't want to reply, he or she can simply close the form. He or she can also click the Delete button. By doing so, this piece of event will be deleted from his or her calendar and from the 'Message list' table. The response (Accept, Tentative or Decline) will update the 'Attendee availability' page and the 'Message list' page automatically.

The user can also set an event as recurrent event. By clicking on the Recurrence button that located on the top of the appointment or meeting scheduling pages (Appendix p.4, 6-7), the recurrence form will be opened, as shown in Appendix p.5. The user can set up various recurrent patterns for the appointment or meeting from here. Four major patterns are available: Daily, Weekly, Monthly and Yearly. The user can also set the range of recurrence, i.e., the starting and ending date of the recurrence. The appointment or meeting time can also be setup in this page.

Security and Database Design

As a security measurement, Users are required to login to use the web calendar. The login screen looks like Fig.7. Form authentication method [9] will be used to verify the user name and password provided by the user. The user's password will be encrypted and saved in the database.

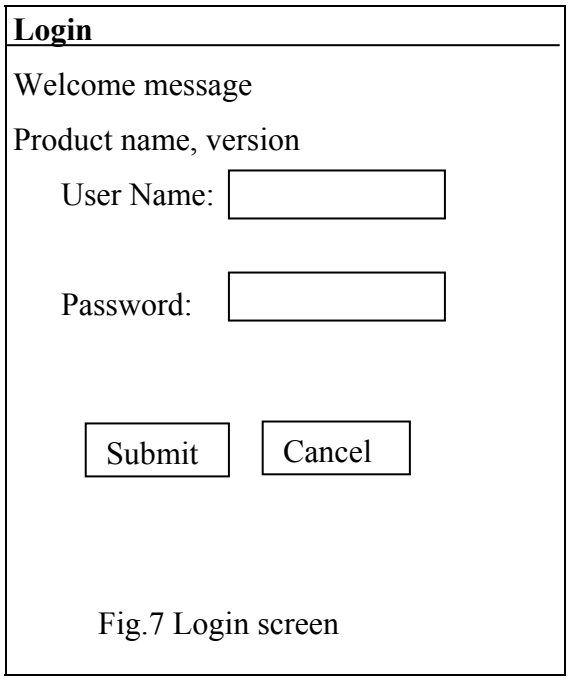


Fig.7 Login screen

When a client's browser accesses the website of the calendar server (for instance, *https://www.mybcalendar.com/login.aspx*), a login screen page will be sent to the client's browser. Then the user needs to submit a valid user name and password to the web server. If login succeeds, a default calendar page (Appendix p.1) will be sent to the client's browser. It is a DHTML page and includes the user's own meeting and appointment schedules. If login fails, the fields of User name and Password will be cleared and wait for re-login.

Nine major user tables will be created in the calendar database. SQL server authentication is used for secure access to the SQL database. When a connection request is made, SQL server receives both a login ID and a password that must be included in the connection string, which it validates against the list of logins identified to it by the system administrator. Samples of these related tables are listed below:

User										
UserID	FirstName	LastName	MiddleName	Email	BPhone	HPhone	BFax	Mobile	UserName	Password
1	Jane	Zhong	j	jane.zhong0@lycos.com	4255366	5867310	5867310		jzhong	D9-54-7B-
2	Jing	Li		jingli@me.rochester.edu	2655443	5867310			jli	DC-55-R-

This table is used for recording all the information of each user (identified by UserID).

Event												
EventID	UserID	Subject	Location	StartDate	StartTime	EndDate	EndTime	ShowTimeAs	Private	Recurrence	Note	Meeting
1	2	See …	Strong	5/2/2004	10:30am	5/2/2004	12:00am	Busy	T	F		F
2	1	Meeting	Room1	3/5/2004	11:00am	5/2/2004	11:30am	Busy	F	T		T

This table is used to record all the event information. Each event is identified by EventID. UserID is used to identify the user who created this event. UserID is also acted as a foreign key which will relate to the User table. Subject, Location, StartDate, StartTime, EndDate, EndTime and Note are used to record the information of the event. ShowTimeAs is used to show others your availability in this time period when they plan meetings. Private is used to show that this event is private or not. If Private is true, this event is not allowed to be viewed by other people. If Recurrence is true, this event is a recurrence event, and the recurrence information will be retrieved from Recurrence table. If Meeting is true, that means this is a meeting event, and the attendee information will be retrieved from MeetingAttendee table.

MeetingAttendee			
EventID	AttendeeUserID	Status	Response
13	1	meeting organizer	None
13	2	Required	Accepted
13	3	Optional	Tentative

This table is used for the meeting event. AttendeeUserID is used to identify the people who will be attended to the meeting. EventID is used to retrieve the information from the Event Table, and is also used to retrieve all the attendee ID for a specific Meeting (identified by the EventID). Status is used to show the attendee is “Required” or “Optional”. Response is used to record the response from the attendee.

Recurrence				
EventID	StartDate	RangePattern	RangeValue	RecurPattern
20	3/5/2004	No End		FreqRecur
13	3/6/2004	End Until	5/6/2005	WkdayRecur

This table is used to record the recurrence event’s information. If the Recurrence field value for an EventID is true in Event table, the recurrence information will be retrieved from this table by the condition of Recurrence.EventID=Event.EventID.

StartDate is the day from which this event starts to be a recurrence event. RangePattern and RangeValue is used to show how long this recurrence will last.

FreqRecurPatn		
EventID	**Frequency**	**RepeatRange**
20	Every Third	Month
11	Every other	Week

This table is used as connection with the table of Recurrence to record the value when RecurPattern value in Recurrence table equals to "FreqRecur".

WkdayRecurPatn			
EventID	**WeekNumber**	**WeekdayValue**	**RepeatRange**
13	Second	Mon	3 month
11	First	Wed	Month

This table is used as connection with the table of Recurrence to record the value when RecurPattern value in Recurrence table equals to "Wkdayrecur".

Proxy		
AssigningUserID	**AssignedUserID**	**Permission**
1	2	r
1	3	rw
3	1	rw
3	2	r

This table is used for proxy. AssigningUserID is used to identify the user who assigns the proxy to other users. AssignedUserID is used to identify the users who is assigned the proxy with Read or Read/Write Permission by another user.

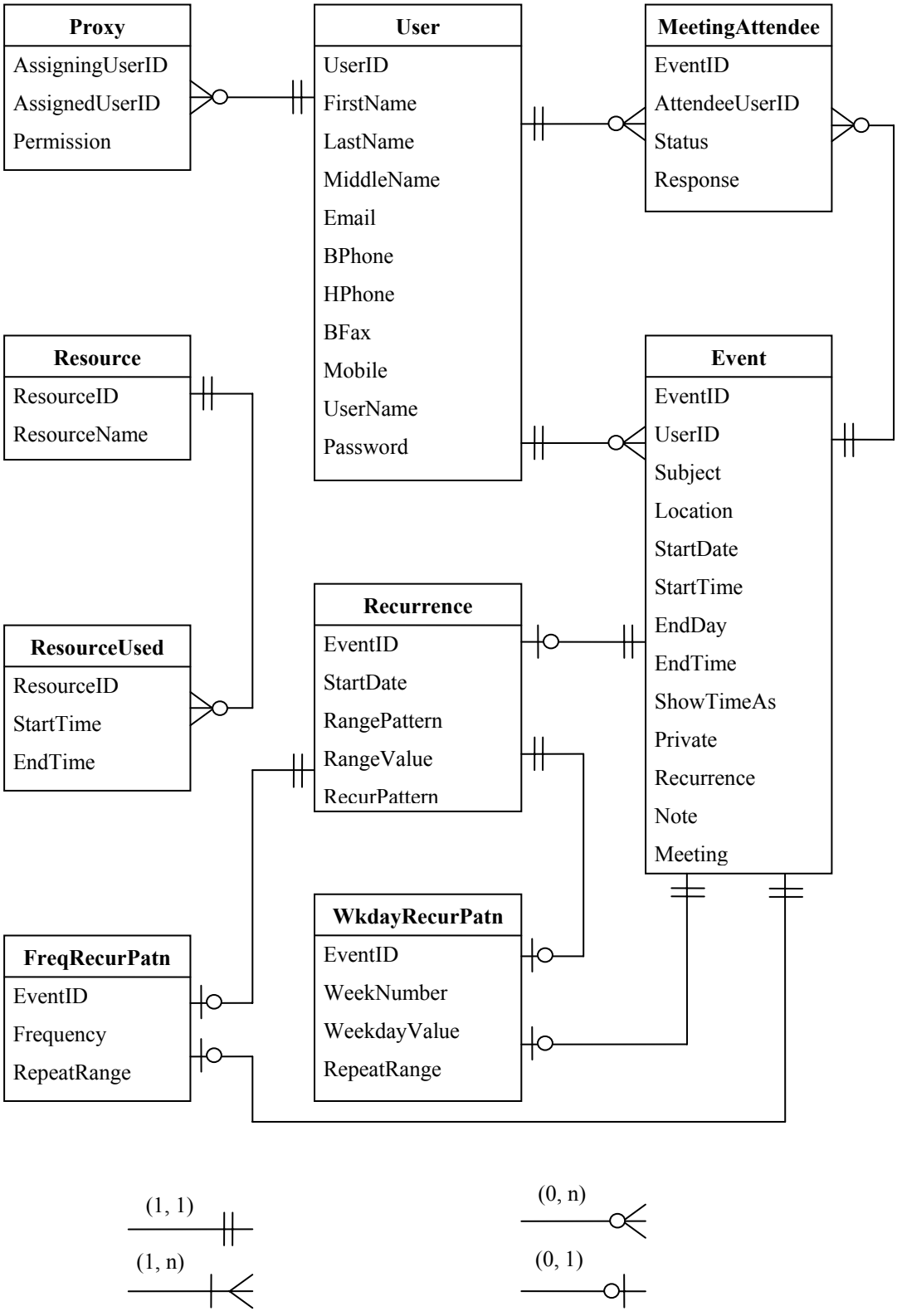
Resource	
ResourceID	**ResourceName**
1	Conference Room1
2	Project
3	Laptop

This table is used for recording all the resource, like the conference room and project, etc..

ResourceUsed		
ResourceID	**StartTime**	**EndTime**
1	3/5/2004 10:00am	3/5/2004 12:00am
3	5/4/2004 9:30am	5/4/2004 11:00am

This table is used for reserving the resource for meeting event.

The Database Entity-Relationship Diagram:



The calendar information is stored in these nine major tables that can be accessed by stored procedures using SQL language. There will be three layers to implement this project. The presentation layer will include all the ASP pages and their code behind files. The business layer will be responsible for access the database and some relative classes. The database layer will implement all the stored procedures using SQL language.

Principle Deliverables

1. A functional web calendar server.
2. Comparison analysis with other kind of calendars, focusing on the functionality, effectives and the interface with feel and look.
3. User manual.
4. Source code and deployment package.
5. Testing examples or demonstration.

References

[1] University of Cincinnati, http://www.ucit.uc.edu/documentation/calendar.pdf, *Scheduling with the Outlook Calendar*, 2002.

[2] Apple Computer Inc., http://www.apple.com/ical/, *iCal Help*, ver.1.5, 2004.

[3] Meeting Maker, Inc., *Meetingmaker User Guide*, p.17, 2001.

[4] Yahoo! Inc., http://www.yahoo.com/, 2004.

[5] MSDN Microsoft Developer Network, *Microsoft .NET Developer Training*, Part No: X08-43203, p.05.4, 2001.

[6] Microsoft Corp., online help in *MSDN Library .NET 2003*.

[7] Farhan Muhammad and Matt Milner, The Age of ASP.NET, *Real World ASP.NET Best Practices*, Apress, p.1, 2003.

[8] Chris Payne, Secure You ASP.NET Applications, *SAMs Teach Yourself ASP.NET in 21 Days,* p.698, 2002.

Provisional Schedule

April 2004	The first version of the proposal was done.
May 2004	The final version of the proposal was done.
Aug. 2004	Finish the project coding and the project test.
Sept. 2004	Finish the project report and defense.

Appendix

See file: Appendix7.ppt.

P.14 P.4 P.10 Display today's calendar P.8 P.2

Add appointment	Add meeting	Message	Go to Today	Contacts	Proxy

P.1 P.3

By Day	By Month

Today is: **January 24, 2004**

Current month

Last month

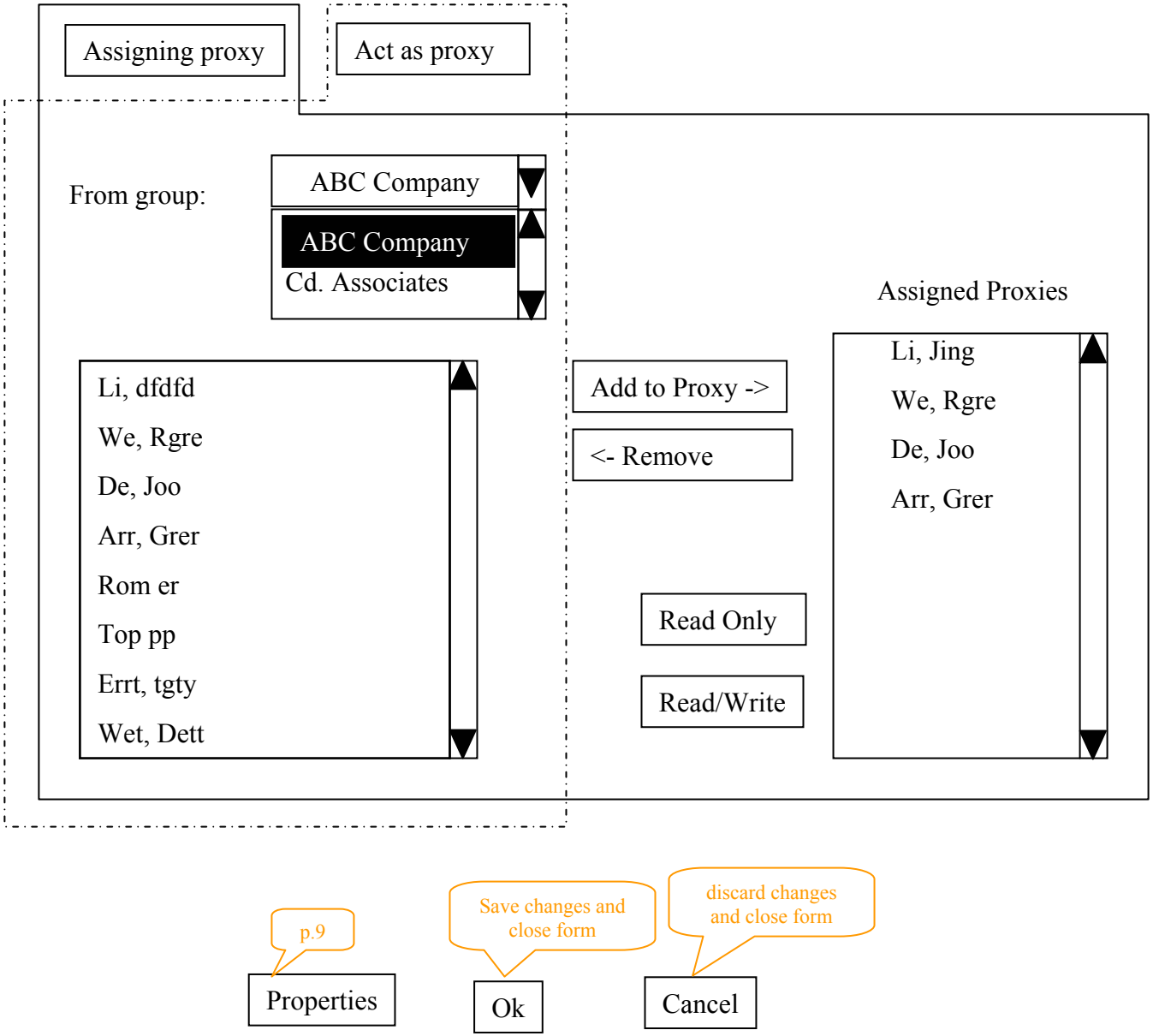
Next month

January 2004						
Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Go to that day

Monday, February 12, 2004

8 :	00	Group Meeting (Build 147-14)
	30	
9 :	00	
	30	
10 :	00	P.4
	30	Seminar (220-23)
11 :	00	
	30	
12 :	00	
	30	
1 :	00	
	30	
2 :	00	
	30	
3 :	00	
	30	
4 :	00	
	30	
5 :	00	
	30	
6 :	00	
	30	Put out garbage bin.
7 :	00	
	30	


Assigning proxy
Act as proxy
From group:
ABC Company
ABC Company
Cd. Associates
Assigned Proxies
Li, dfdfd
We, Rgre
De, Joo
Arr, Grer
Rom er
Top pp
Errt, tgty
Wet, Dett
Add to Proxy ->
<- Remove
Li, Jing
We, Rgre
De, Joo
Arr, Grer
Read Only
Read/Write
p.9
Properties
Save changes and close form
Ok
discard changes and close form
Cancel

Add appointment	Add meeting	Message	Go to Today	Contacts	Proxy

Today is: **January 24, 2004**

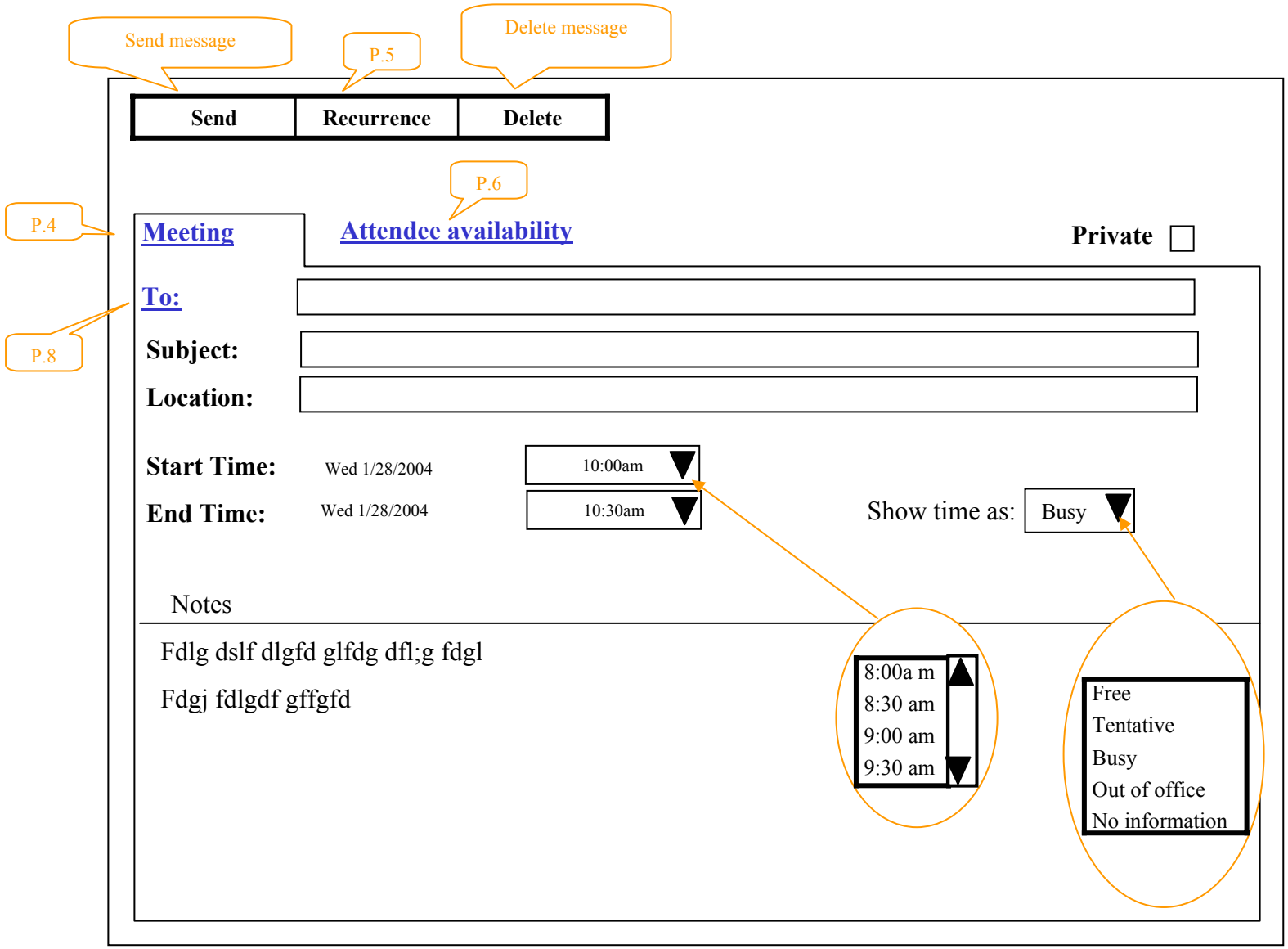
By Day	By Month

January 2004

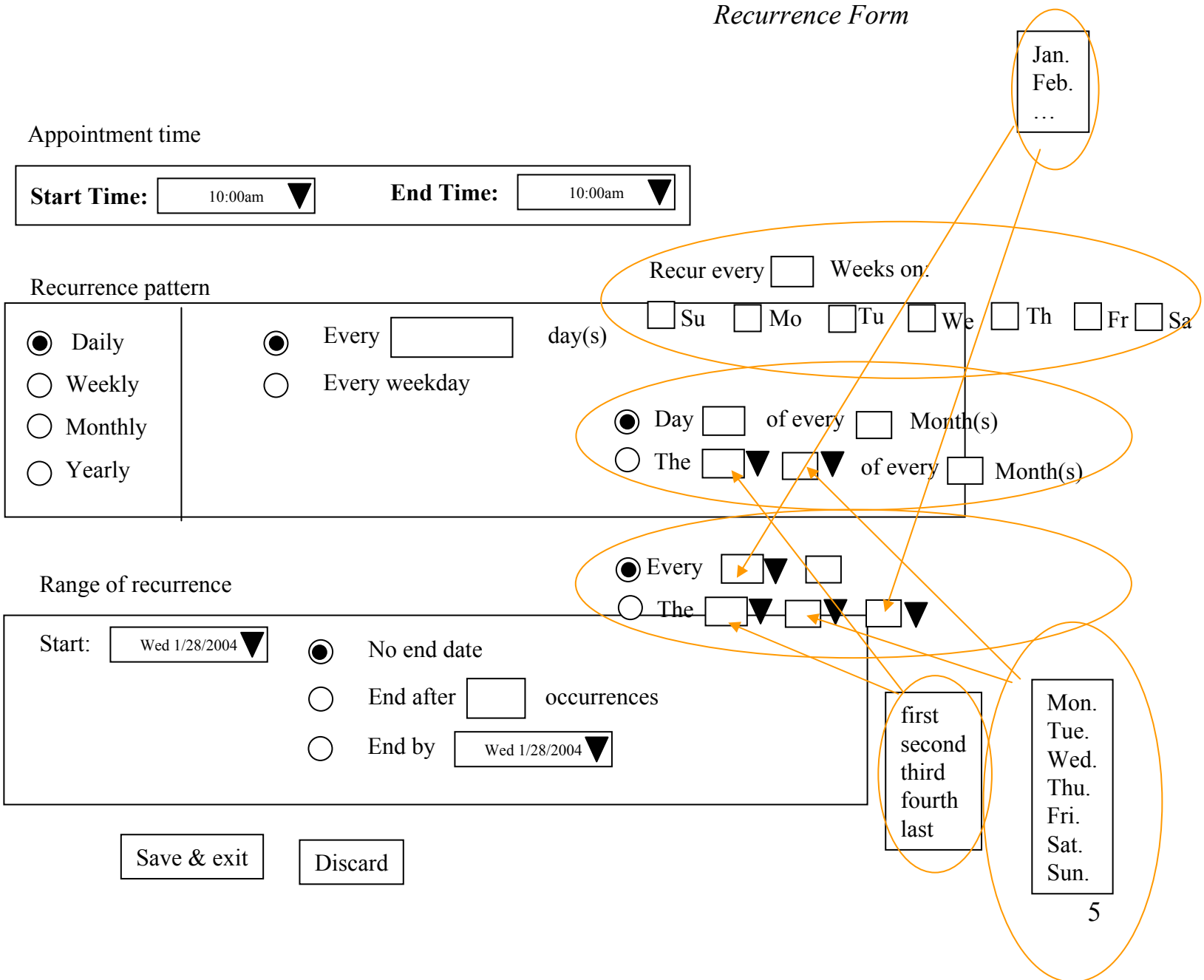
Su	Mo	Tu	We	Th	Fr	Sa
			P.1	1	2	3
4	5 Tret re P.4 rtrt rtrete Gh ghg gfhg	6	7	8	9	10
11	12	13	14 Fd dsf dsc P.4 Gbg fgfgrg fgfggfg	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

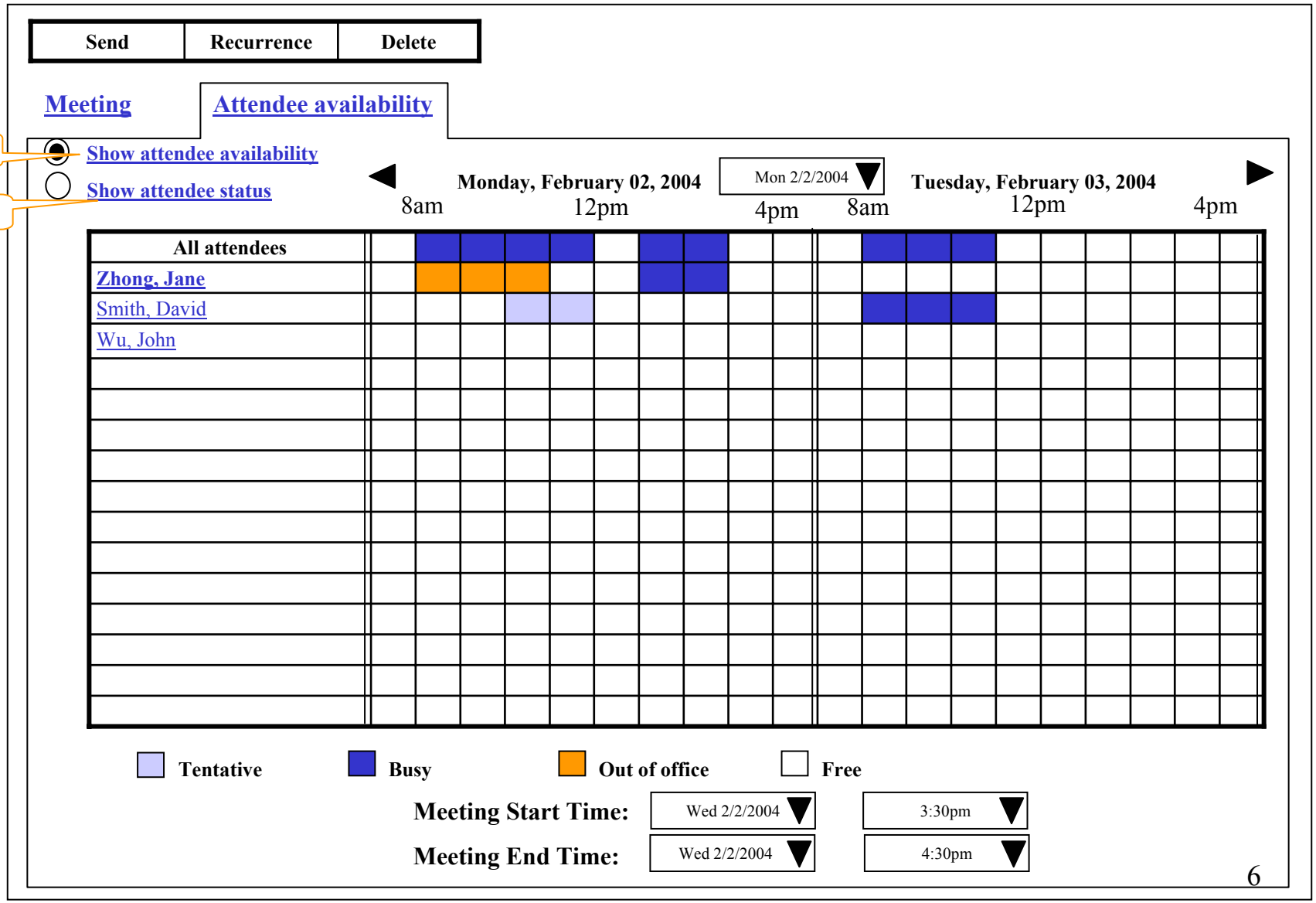
January 2004

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31


Send message
P.5
Delete message
Send
Recurrence
Delete
P.6
P.4
Meeting
Attendee availability
Private
To:
P.8
Subject:
Location:
Start Time:
Wed 1/28/2004
10:00am
End Time:
Wed 1/28/2004
10:30am
Show time as:
Busy
Notes
Fdlg dslf dlgfd glfdg dfl;g fdgl
Fdgj fdlgdf gffgfd
8:00a m
8:30 am
9:00 am
9:30 am
Free
Tentative
Busy
Out of office
No information

Recurrence Form



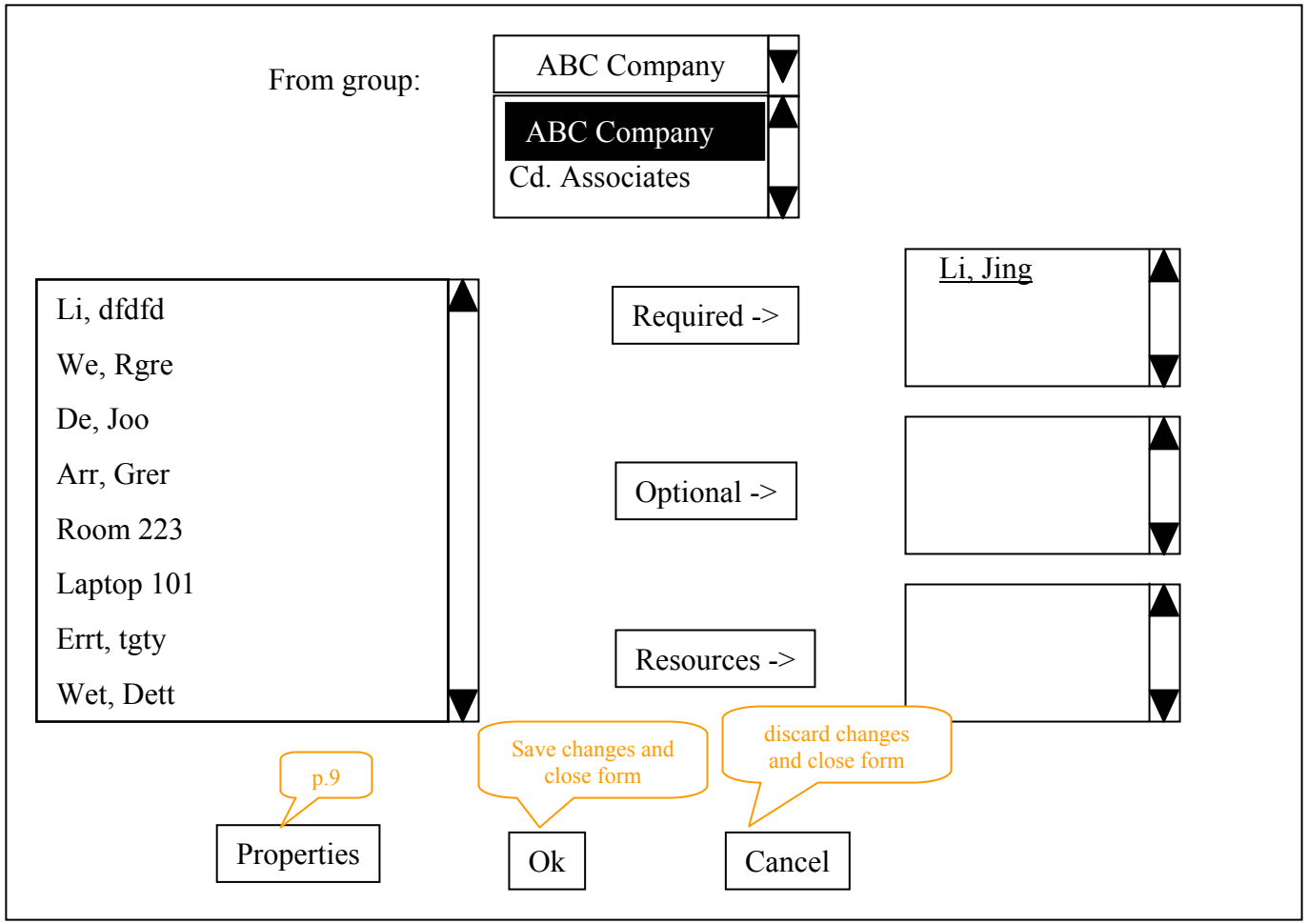

Send
Recurrence
Delete
Meeting
Attendee availability
P.6
Show attendee availability
Show attendee status
P.7
Monday, February 02, 2004
Mon 2/2/2004
Tuesday, February 03, 2004
8am
12pm
4pm
8am
12pm
4pm
All attendees
Zhong, Jane
Smith, David
Wu, John
Tentative
Busy
Out of office
Free
Meeting Start Time:
Wed 2/2/2004
3:30pm
Meeting End Time:
Wed 2/2/2004
4:30pm

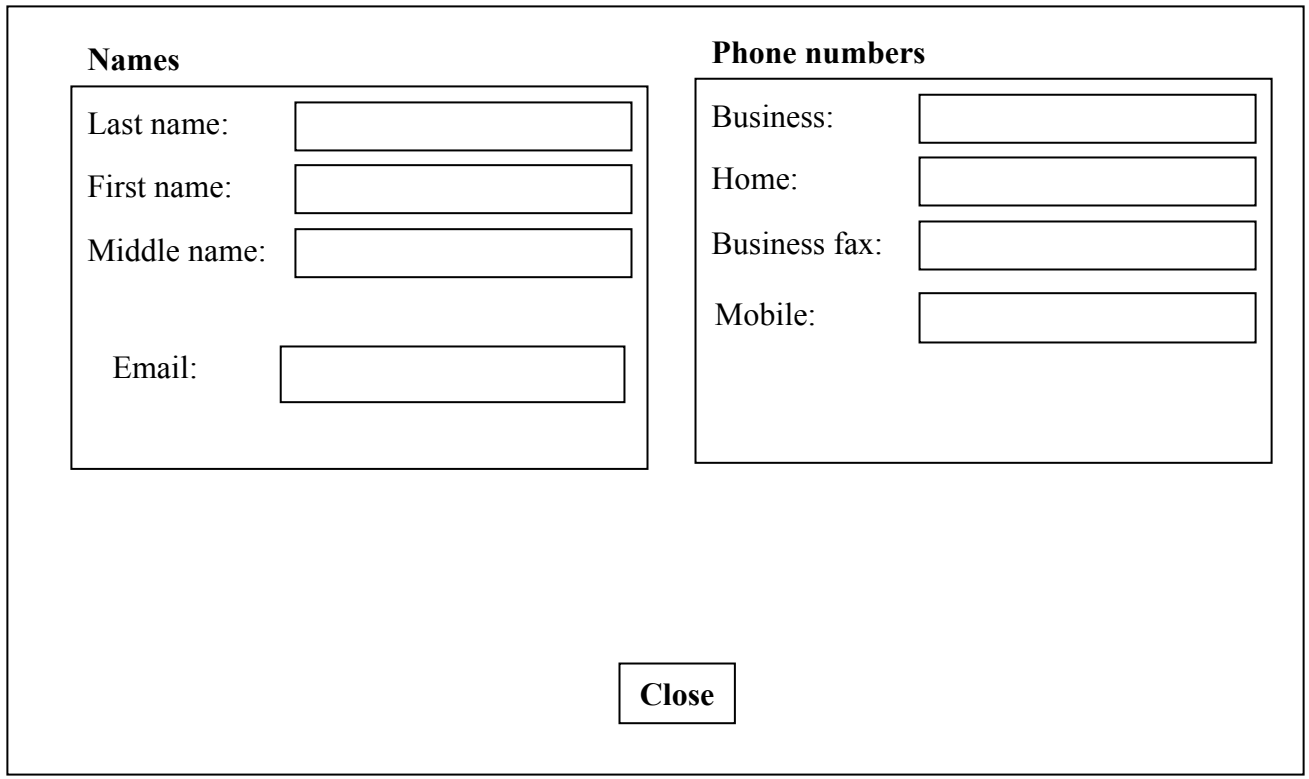
Send	Recurrence	Delete

Meeting | Attendee availability

- ○ Show attendee availability
- ◉ Show attendee status

Name	Attendee	Response
Zhong, Jane	**Meeting organizer**	**None**
Smith, David	Required	**Accepted**
Wu, John	Optional	**Declined**
Room 204	Resource	**Accepted**
Tee, John	Required	**Tentative**


From group:
ABC Company
ABC Company
Cd. Associates
Li, dfdfd
We, Rgre
De, Joo
Arr, Grer
Room 223
Laptop 101
Errt, tgty
Wet, Dett
Required ->
Optional ->
Resources ->
Li, Jing
p.9
Properties
Save changes and close form
Ok
discard changes and close form
Cancel

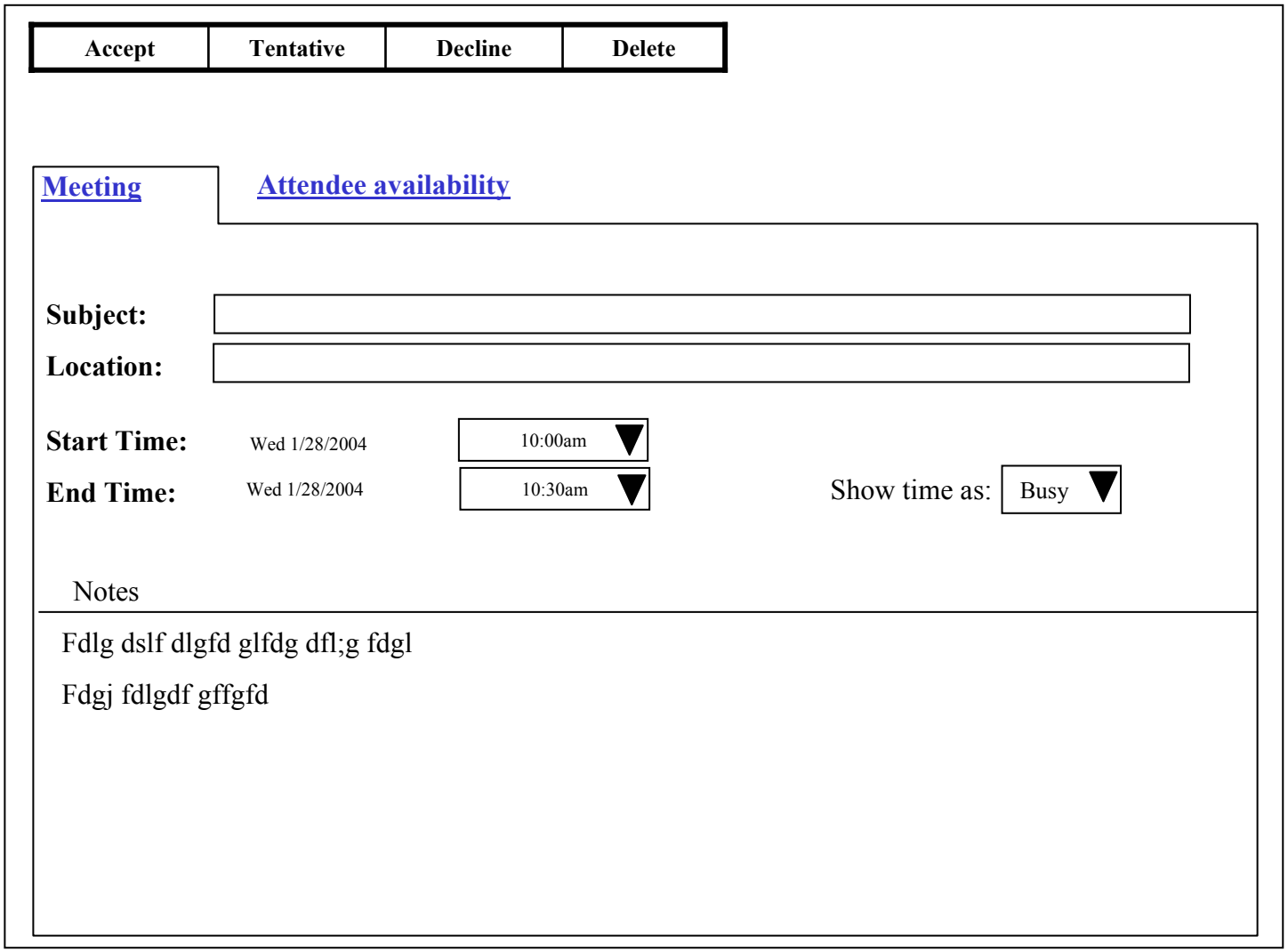

Names
Last name:
First name:
Middle name:
Email:
Phone numbers
Business:
Home:
Business fax:
Mobile:
Close

Message list:

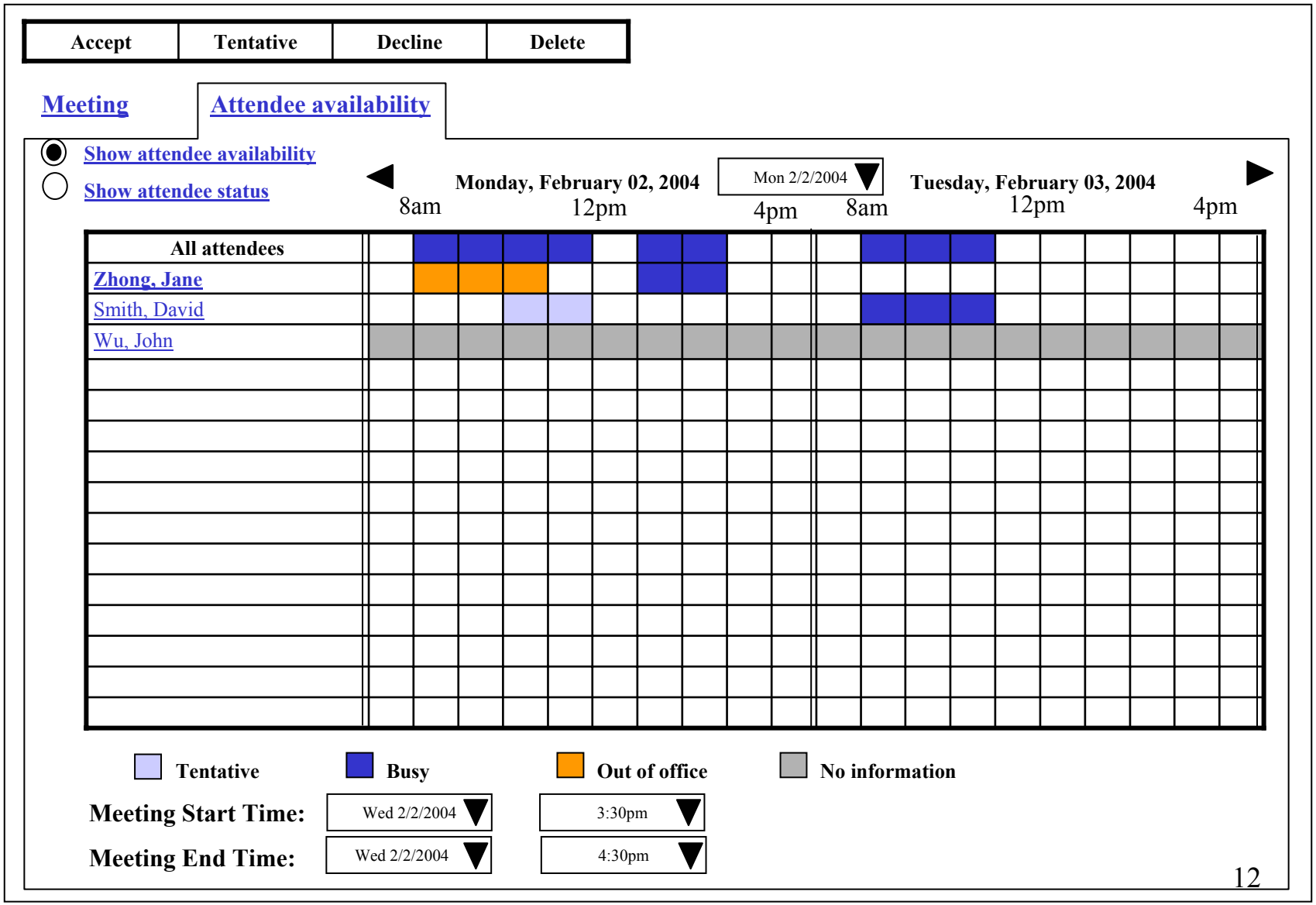
Senders	Subject	Location	Date	Status
Li, Teresa	Go fishing	Dfd s	Tue, 1/29/2004	Accept
Race, Tom	Discuss …	Ghjh gfht	Wed, 1/30/2004	

p.11

Read only mode



Read only mode



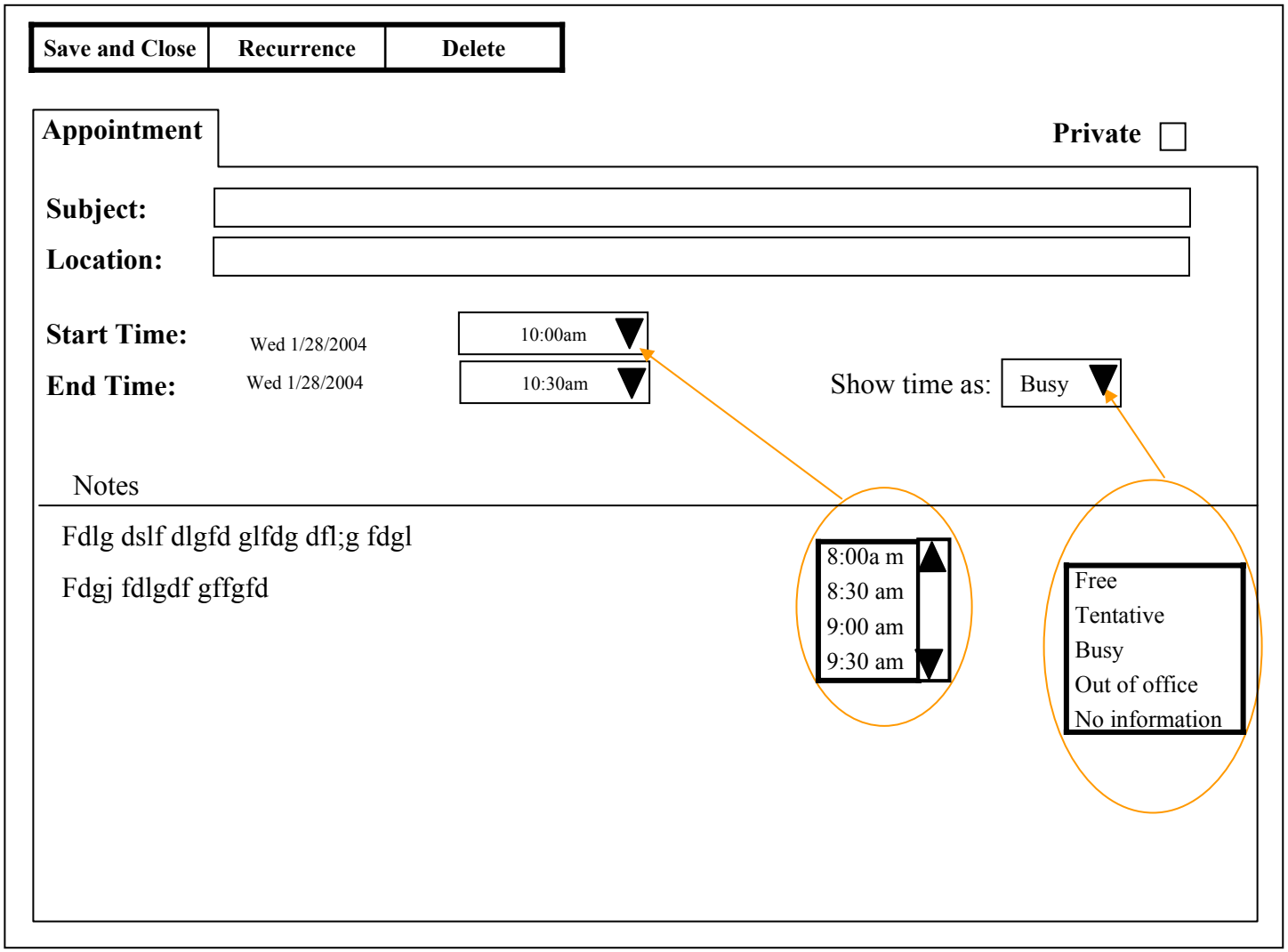
Read only mode

Accept	Tentative	Decline	Delete

Meeting | Attendee availability

○ Show attendee availability
◉ Show attendee status

Name	Attendee	Response
Zhong, Jane	**Meeting organizer**	**None**
Smith, David	Required	**Accepted**
Wu, John	Optional	**Declined**
Room 204	Resource	**Accepted**
Tee, John	Required	**Tentative**


Save and Close
Recurrence
Delete
Appointment
Private
Subject:
Location:
Start Time:
Wed 1/28/2004
10:00am
End Time:
Wed 1/28/2004
10:30am
Show time as:
Busy
Notes
Fdlg dslf dlgfd glfdg dfl;g fdgl
Fdgj fdlgdf gffgfd
8:00a m
8:30 am
9:00 am
9:30 am
Free
Tentative
Busy
Out of office
No information